



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

invariant address range

SEARCH

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used invariant address range

Found 38,933 of 148,786

Sort results
by

relevance

[Save results to a Binder](#)Try an [Advanced Search](#)Try this search in [The ACM Guide](#)Display
results

expanded form

[Search Tips](#)
☐ Open results in a new
window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [New architectures: Invariants: a new design methodology for network architectures](#)

Bengt Ahlgren, Marcus Brunner, Lars Eggert, Robert Hancock, Stefan Schmid

 August 2004 **Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture**

 Full text available: [pdf\(230.28 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The first age of Internet architectural thinking concentrated on defining the correct principles for designing a packet-switched network and its application protocol suites. Although these same principles remain valid today, they do not address the question of how to reason about the evolution of the Internet or its interworking with other networks of very different heritages. This paper proposes a complementary methodology, motivated by the view that evolution and interworking flexibility are d ...

Keywords: design methodology, design principles, invariants, network architecture design

2 [Optimization of array subscript range checks](#)

Jonathan M. Asuru

 June 1992 **ACM Letters on Programming Languages and Systems (LOPLAS)**, Volume 1
Issue 2

 Full text available: [pdf\(619.54 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Compile-time elimination of subscript range checks is performed by some optimizing compilers to reduce the overhead associated with manipulating array data structures. Elimination and propagation, the two methods of subscript range check optimization, are less effective for eliminating global redundancies especially in while-loop structures with nonconstant loop guards. This paper describes a subscript range check optimization procedure that can eliminate more range checks than current meth ...

Keywords: conservative expression substitution, loop guard elimination, range check optimization

3 [Practical data breakpoints: design and implementation](#)

Robert Wahbe, Steven Lucco, Susan L. Graham

 June 1993 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1993 conference on Programming language design and implementation**, Volume 28 Issue 6

Additional Information:


Full text available:  [pdf\(1.37 MB\)](#)

[full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A data breakpoint associates debugging actions with programmer-specified conditions on the memory state of an executing program. Data breakpoints provide a means for discovering program bugs that are tedious or impossible to isolate using control breakpoints alone. In practice, programmers rarely use data breakpoints, because they are either unimplemented or prohibitively slow in available debugging software. In this paper, we present the design and implementation of a practical data breakpoint ...

4 [Catching bugs in the web of program invariants](#)

Cormac Flanagan, Matthew Flatt, Shriram Krishnamurthi, Stephanie Weirich, Matthias Felleisen
May 1996 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1996 conference on Programming language design and implementation**, Volume 31 Issue 5

Full text available:  [pdf\(1.14 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

MrSpidey is a user-friendly, interactive static debugger for Scheme. A static debugger supplements the standard debugger by analyzing the program and pinpointing those program operations that may cause run-time errors such as dereferencing the null pointer or applying non-functions. The program analysis of MrSpidey computes value set descriptions for each term in the program and constructs a value flow graph connecting the set descriptions. Using the set descriptions, MrSpidey can identify and handle ...

5 [A formal framework for modelling and analysing mobile systems](#)

Graeme Smith
January 2004 **Proceedings of the 27th conference on Australasian computer science - Volume 26**

Full text available:  [pdf\(171.04 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#)

This paper presents a formal framework for modelling and analysing mobile systems. The framework comprises a collection of models of the dominant design paradigms which are readily extended to incorporate details of particular technologies, i.e., programming languages and their run-time support, and applications. The modelling language is Object-Z, an extension of the well-known Z specification language with explicit support for object-oriented concepts. Its support for object orientation makes ...

Keywords: formal methods, mobile computing, object orientation

6 [Scalable high-speed prefix matching](#)

Marcel Waldvogel, George Varghese, Jon Turner, Bernhard Plattner
November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4

Full text available:  [pdf\(933.02 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Finding the longest matching prefix from a database of keywords is an old problem with a number of applications, ranging from dictionary searches to advanced memory management to computational geometry. But perhaps today's most frequent best matching prefix lookups occur in the Internet, when forwarding packets from router to router. Internet traffic volume and link speeds are rapidly increasing; at the same time, a growing user population is increasing the size of routing tables against which ...

Keywords: collision resolution, forwarding lookups, high-speed networking

7

[Face recognition: A literature survey](#)

W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld
December 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 4

Full text available:  [pdf\(4.28 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


As one of the most successful applications of image analysis and understanding, face recognition has recently received significant attention, especially during the past several years. At least two reasons account for this trend: the first is the wide range of commercial and law enforcement applications, and the second is the availability of feasible technologies after 30 years of research. Even though current machine recognition systems have reached a certain level of maturity, their success is ...

Keywords: Face recognition, person identification

8 Control flow analysis in scheme

O. Shivers

June 1988 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1988 conference on Programming Language design and Implementation**, Volume 23 Issue 7

Full text available:  [pdf\(1.10 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Traditional flow analysis techniques, such as the ones typically employed by optimizing Fortran compilers, do not work for Scheme-like languages. This paper presents a flow analysis technique — control flow analysis — which is applicable to Scheme-like languages. As a demonstration application, the information gathered by control flow analysis is used to perform a traditional flow analysis problem, induction variable elimination. Extensions and limitations are discussed ...

9 Global array reference allocation

Guido Araujo, Guilherme Ottoni, Marcelo Cintra

April 2002 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 7 Issue 2

Full text available:  [pdf\(354.75 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Embedded systems executing specialized programs have been increasingly responsible for a large share of the computer systems manufactured every year. This trend has increased the demand for processors that can guarantee high-performance under stringent cost, power, and code size constraints. Indirect addressing is by far the most used addressing mode in programs running on these systems, since it enables the design of small and faster instructions. This paper proposes a solution to the problem of ...

Keywords: Address registers, DSPs, auto-increment addressing modes, register allocation

10 Three-dimensional object recognition

Paul J. Besl, Ramesh C. Jain

March 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 1


Full text available:  [pdf\(7.76 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

A general-purpose computer vision system must be capable of recognizing three-dimensional (3-D) objects. This paper proposes a precise definition of the 3-D object recognition problem, discusses basic concepts associated with this problem, and reviews the relevant literature. Because range images (or depth maps) are often used as sensor input instead of intensity images, techniques for obtaining, processing, and characterizing range data are also surveyed.

11 Why we don't know how to simulate the Internet

Vern Paxson, Sally Floyd

December 1997 **Proceedings of the 29th conference on Winter simulation**

Full text available:  pdf(1.04 MB)


Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



12 Image retrieval by appearance

S. Ravela, R. Manmatha

July 1997 **ACM SIGIR Forum , Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval**, Volume 31 Issue SI

Full text available:  pdf(1.47 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



13 Debugging highly-optimized Ada with code motion (DHACM)

Kevin Tucker

November 1997 **Proceedings of the conference on TRI-Ada '97**

Full text available:  pdf(721.40 KB)


Additional Information: [full citation](#), [references](#), [index terms](#)



14 Scalable high speed IP routing lookups

Marcel Waldvogel, George Varghese, Jon Turner, Bernhard Plattner

October 1997 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication**, Volume 27 Issue 4

Full text available:  pdf(1.66 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)




Internet address lookup is a challenging problem because of increasing routing table sizes, increased traffic, higher speed links, and the migration to 128 bit IPv6 addresses. IP routing lookup requires computing the best matching prefix, for which standard solutions like hashing were believed to be inapplicable. The best existing solution we know of, BSD radix tries, scales badly as IP moves to 128 bit addresses. Our paper describes a new algorithm for best matching prefix using binary search o ...

15 Low-synchronization translation lookaside buffer consistency in large-scale shared-memory multiprocessors

B. Rosenberg

November 1989 **ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium on Operating systems principles**, Volume 23 Issue 5

Full text available:  pdf(1.08 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)



Operating systems for most current shared-memory multiprocessors must maintain translation lookaside buffer (TLB) consistency across processors. A processor that changes a shared page table must flush outdated mapping information from its own TLB, and it must force the other processors using the page table to do so as well. Published algorithms for maintaining TLB consistency on some popular commercial multiprocessors incur excessively high synchronization costs. We present an efficient TLB ...


16 Web server workload characterization: the search for invariants

Martin F. Arlitt, Carey L. Williamson

May 1996 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1996**



ACM SIGMETRICS international conference on Measurement and modeling of computer systems, Volume 24 Issue 1


Full text available:  [pdf\(1.28 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The phenomenal growth in popularity of the World Wide Web (WWW, or the Web) has made WWW traffic the largest contributor to packet and byte traffic on the NSFNET backbone. This growth has triggered recent research aimed at reducing the volume of network traffic produced by Web clients and servers, by using caching, and reducing the latency for WWW users, by using improved protocols for Web interaction. Fundamental to the goal of improving WWW performance is an understanding of WWW workloads. This ...

17 Stride prefetching by dynamically inspecting objects

Tatsushi Inagaki, Tamiya Onodera, Hideaki Komatsu, Toshio Nakatani

May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation**, Volume 38 Issue 5

Full text available:  [pdf\(168.64 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Software prefetching is a promising technique to hide cache miss latencies, but it remains challenging to effectively prefetch pointer-based data structures because obtaining the memory address to be prefetched requires pointer dereferences. The recently proposed stride prefetching overcomes this problem, but it only exploits *inter-iteration* stride patterns and relies on an off-line profiling method. We propose a new algorithm for stride prefetching which is intended for use in a dynamic ...

Keywords: Java just-in-time compiler, object inspection, stride prefetching

18 A structural view of the Cedar programming environment

Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann

August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4

Full text available:  [pdf\(6.32 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

19 Efficient software-based fault isolation

Robert Wahbe, Steven Lucco, Thomas E. Anderson, Susan L. Graham

December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles**, Volume 27 Issue 5

Full text available:  [pdf\(1.49 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

One way to provide fault isolation among cooperating software modules is to place each in its own address space. However, for tightly-coupled modules, this solution incurs prohibitive context switch overhead. In this paper, we present a software approach to implementing fault isolation within a single address space. Our approach has two parts. First, we load the code and data for a distrusted module into its own *fault domain*, a logically separate portion of the application's address space ...

20 Effective exploitation of a zero overhead loop buffer

Gang-Ryung Uh, Yuhong Wang, David Whalley, Sanjay Jinturkar, Chris Burns, Vincent Cao
May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 workshop
on Languages, compilers, and tools for embedded systems**, Volume 34 Issue 7

Full text available:  [pdf\(1.01 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A Zero Overhead Loop Buffer (ZOLB) is an architectural feature that is commonly found in DSP processors. This buffer can be viewed as a compiler managed cache that contains a sequence of instructions that will be executed a specified number of times. Unlike techniques such as loop unrolling, a loop buffer is a hardware technique that can be used to minimize loop overhead without the penalty of increasing code size. In addition, a ZOLB also requires relatively little space and power, which are bo ...

Results 1 - 20 of 200

Result page: **1** [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

Refine Search

Search Results -

Terms	Documents
L6 AND compile	14

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Search:

L7

Search History

 DATE: Monday, January 17, 2005 [Printable Copy](#) [Create Case](#)
Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=NO; OP=OR

<u>L7</u>	L6 AND compile	14	<u>L7</u>
<u>L6</u>	L5 AND (lower ADJ bound)	73	<u>L6</u>
<u>L5</u>	L4 AND (upper ADJ bound)	120	<u>L5</u>
<u>L4</u>	L2 AND (bound)	665	<u>L4</u>
<u>L3</u>	L2 AND interned	0	<u>L3</u>
<u>L2</u>	L1 AND (address AND range)	2146	<u>L2</u>
<u>L1</u>	invariant	12722	<u>L1</u>

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L6 AND (7\$\$\$/\$\$\$\$.ccls.)	36

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Search:

L9

Search History

DATE: Monday, January 17, 2005 [Printable Copy](#) [Create Case](#)

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=NO; OP=OR

<u>L9</u>	L6 AND (7\$\$\$/\$\$\$\$.ccls.)	36	<u>L9</u>
<u>L8</u>	L5 AND (compile near invariant)	0	<u>L8</u>
<u>L7</u>	L6 AND compile	14	<u>L7</u>
<u>L6</u>	L5 AND (lower ADJ bound)	73	<u>L6</u>
<u>L5</u>	L4 AND (upper ADJ bound)	120	<u>L5</u>
<u>L4</u>	L2 AND (bound)	665	<u>L4</u>
<u>L3</u>	L2 AND interned	0	<u>L3</u>
<u>L2</u>	L1 AND (address AND range)	2146	<u>L2</u>
<u>L1</u>	invariant	12722	<u>L1</u>

END OF SEARCH HISTORY